全球資訊網中網頁-動作使用路徑的資料挖掘

楊亨利
政治大學資訊管理學系

林青峰
政治大學資訊管理學系

摘要

客戶在從事消費時，往往會有許多不一樣的行為產生。對組織而言，研究客戶的消費行為能夠協助組織更了解客戶的資訊，進而支援其經營活動。以往與客戶行為相關的資料挖掘研究，較著重於客戶的消費資料。而對於客戶在商店中做了那些動作，及其動作會導致發生的事件並沒有較全盤及深入的討論。對實體業者而言，要實際的去記錄使用者在商店內的行為，是不太可行的；但從另一個層面來說，隨著網際網路與資料收集技術的發展，網站經營者應用 log 留存技術，將比傳統業者更容易且完整的收集到消費者行為記錄。本研究試圖在全球資訊網的環境中建立一個能夠同時分析使用者的瀏覽網頁路徑及其動作過程的演算法；並且配合該演算法建置一個雛形系統，以驗証其效能，最後並評估其日後實務操作的可行性。

關鍵字：全球資訊網、網站使用挖掘、資料挖掘、使用者行為、網頁路徑、動作路徑

Data Mining of Page and Action Path on Web

Heng-Li Yang
MIS Department, National Cheng-Chi University

Qing-Fung Lin
MIS Department National Cheng-Chi University

ABSTRACT

The purchasing behaviors of different customers are various. The study of customer purchasing behavior can help organizations understand their client intentions to support their business activities. In the past, customer behavior data mining emphasized on the purchase items, i.e., what customers buy. There were few studies discussing what paths they took and what actions they made in an e-store. It is impossible for a physical store to record its customers' all actions and passing paths. However, a website store can easily collect such data in an Internet log. This study proposes a data mining algorithm that can analyze both customers' browsing pages and their actions paths. The algorithm's efficiency and feasibility were examined in our prototype. This study would contribute to help the website mangers to restructure their website layouts or advertisement position to catch the customer's eyes.

KeyWords: World-Wide-Web; Web-Usage Mining; Data-Mining; User Behavior; Page Path; Action Path

壹、研究背景與目的

在激烈的商業競爭中，行銷策略的制訂無疑是影響組織能否達成獲利目標很重要的一個環節。「瞭解客戶行為」又是組織想制訂良好行銷策略時首先應該擁有的資訊。無論是個人或是企業，如果能比競爭者更正確且有效率的從其所擁有的大量資料中找出更多未知的知識或資訊，自然在競爭場上就處於優勢的位置。資料挖掘(Data Mining)，或稱為知識發現(Knowledge Discovery)的研究議題便因為這樣的背景而漸漸的被重視，許多針對不同資料結構的資料發掘方法也被廣泛的討論 (Frawley et al.,1991; Chen, et al., 1996)。

在以往的傳統組織中，往往只能從客戶的消費、服務及維修等日常紀錄中著手收集與客戶有關的行為資訊。當組織想要有系統的收集消費者在商店內的選購行為或選購路徑等進一步的使用者資料時，組織需要面臨付出大量人力與新設備的投資才得以實現。隨著網際網路與資料收集技術的發展，商業網站經營者應用使用者 log 留存的技術，可以很容易的收集到使用者在其網站上的各種行為資料，相較於傳統業者來說，這些資料不僅範圍廣泛，而且十分精確。所以有越來越多的網站經營者開始重視使用者在網站中的行為。在網路泡沫化後，更重視客戶與深入瞭解客戶是他們能夠和傳統業者持續競爭的一個重要利基點。

以往研究全球資訊網中使用者瀏覽行為的文獻，大部份學者皆專注於找出瀏覽頁面路徑部份的規則(Chen,Park and Yu,1998;陳仕昇等，民 88)。我們很少看到文獻會談到使用者在同一網頁中所進行的動作或是跨越網頁間動作路徑的問題。但對在全球資訊網上的使用者而言，使用者其實是藉由執行一個個的動作來控制網頁的路徑。網頁的瀏覽路徑也許可以提供使用者行為在意圖上某一程度的資訊，但也許並不周全。

假設表一為某網頁記錄。對該表第二欄，以一般的網頁路徑技術來看，我們可以找出「若使用者瀏覽過『產品資訊頁』後，接著瀏覽『訂單輸入頁』，則有 50%的機率會完成交易」。但是對該表第三欄的資料，我們可以觀察出「當使用者在『產品資訊頁』中執行過『檢視可搭配種類』後，有 100%的機率在『訂單輸入頁』會『完成交易』」這樣的規則。對於網站的經營者而言，這樣的規則對於更深入了解使用者行為是很有幫助的。本研究便是試圖發展出一種「網頁-動作路徑演算法」來找出其瀏覽行為的關聯模式。

表一：使用者在全球資訊網中可能的網頁與動作路徑

SessionId	一般的網頁路徑	本研究所考慮的網頁動作路徑
A00000001	特價品資訊 → 產品資訊頁 → 訂單輸入頁	特價品資訊[特價資訊查詢→查詢產品明細] → 產品資訊頁[檢視產品明細 → 選擇周邊產品 → 檢視可搭配種類] → 訂單輸入頁[輸入訂購資訊 → 完成下單 → 完成交易]
A00000002	產品資訊頁 → 訂單輸入頁	產品資訊頁[檢視產品明細] → 訂單輸入頁[輸入訂購資訊 → 取消交易]
A00000003	特價品資訊 → 產品資訊頁 → 訂單輸入頁	特價品資訊[特價資訊查詢→查詢產品明細] → 產品資訊頁[檢視產品明細] → 訂單輸入頁[輸入訂購資訊 → 取消交易]
A00000004	產品資訊頁 → 訂單輸入頁	產品資訊頁[檢視產品明細 → 檢視可搭配種類] → 訂單輸入頁[輸入訂購資訊 → 完成下單 → 完成交易]
A00000005	特價品資訊 → 訂單輸入頁	特價品資訊[特價資訊查詢] → 訂單輸入頁[輸入訂購資訊 → 取消交易]

貳、文獻探討

一、資料挖掘

資料挖掘(Data Mining)或被稱為資料庫裏的知識發現(Knowledge Discovery in Databases, KDD)是將隱性的、顯然還不知道且具有有用潛力的資訊經由一種不簡單(Nontrivial)的方式由資料中萃取出來的過程(Frawley et al., 1991) 。資料挖掘演算法的技術經由學者們不斷的發展，已有十分豐富的文獻。各種針對特別資料庫所產生的挖掘演算法也一一的被提出，並且廣泛的被人們所討論(Frawley et al.1991; Agrawal and Srikant,1994;Han and Fu, 1995; Huan, et al., 2003)。

其中 Han 的概念樹演算法是利用各領域專家所建立概念階層(Concept Hierarchy)的資料掘取演算法(Han,et al.,1991; Han, et al., 1992)。而另外與本研究找尋關聯規則相關者，最具代表性的是 Agrawal 與 Skikant 於 1994 年所提的 Apriori 演算法。該演算法可分為兩個部份，第一部份是要在資料庫中找尋出所有的大項目組合(Large Itemset)，其基本精神就是利用掃描資料庫來計算各個候選項目組合(Candidate Itemset)的支援值(Support)。一旦 Support 值超過某個最小 Support 值(Minal Support)，這樣的候選項目組合即可被稱為大的項目組合。再經由 Join 兩個上一階的 Large Itemset，可以產生出下一階的候選項目組合。有了新一階的候選項目組合後，就再重覆的掃描下去，一直到再也找不到新的 Large Itemset 為止。第二部份則是從大項目組合中找出可能的關聯規則。

在 Apriori 演算法中的找尋 Large Itemset 的部份中，往往佔了整個挖掘的大部份的時間。有許多的學者們提出改善的方法；如 Savasere 等人（1995）提出 Partition 的演算法，經由將資料邏輯上切分為幾個不重疊的資料庫，而加強了 Apriori 的效能。Toivonen（1996）使用了隨機抽樣的方法先快速大量的找出 Large Itemset，雖然說在第一輪中有可能會有某些項目會遺失，但是可再進行下一輪的掃描來取得遺失項目。Pasquier 等人（1999）使用最小封閉方格法來處理關聯規則，也成功的加速了 Large Itemset 的尋找過程。Pei 等人(2004)提出 Pattern-Growth 新的作法，以解決大量 Candidate 之問題。

二、網路上的規則挖掘(Web Mining)

隨著近十年來全球資訊網的發展，在 WWW 上的資料挖掘已成為一個引起學者們注意的領域。廣義的來說，Web Mining 可以泛指一切把未知、有用且隱藏在網路中資訊所找出來的活動。一般而言可以分為以下三個種類(Cooley,et al., 1997a)：

(1)針對網路資料內容的挖掘(Web Content Mining)：對於實際存在於網站中的資料當做原料來進行資料發掘。包含以 HTML 語法格式資料的原始網頁資料發掘(Web Page Content Mining)；或是對使用者利用搜尋引擎所得的查詢結果的資料進行的資料發掘(Search Result Mining)。

(2)針對網站的鏈結結構的挖掘(Web Structure Mining)：包括發掘網站間的超鏈接結構的組成，以及網站被別個超鏈接參照間的關係。

(3)針對網站的使用狀況的挖掘(Web Usage Mining)：包括網站伺服器 Log 檔的資料發掘(Chen,Park and Yu,1998;Pei,et al.,2000;陳仕昇等，民 88)，這類的資料挖掘目的在於找出使用者在網站中的瀏覽行為模式，以用來對網站整體的架構進行調整或是調整符合使用者需求的服務 (Batista and Silva, 2002; Baglioni, et al., 2003; Liu, et al., 2003)。

我們的研究在這個分類中是屬於網站的使用狀況的挖掘。對網站使用狀況的挖掘，往往是經營者為了強化其對使用者的瞭解而進行的。所以挖掘的內容都是與使用者行為有關的資料。文獻上常見的是使用者瀏覽網頁路徑的挖掘。在實作 Web Usage Mining 時，一般可以分三個階段(Cooley et al.,1999)：(1)前置處理(Preprocessing)：將可使用的 Log 檔，轉換成可進行演算法處理的資料。(2)演算法(Mining Algorithm)：此為主要核心以找到規則。(3)樣式分析(Pattern Analysis)：在找出規則之後，還要評估找到的樣式是否合理或是有用，並備日後查詢 (Nanopoulos, et al., 2003)。

在實行完這三個階段後，研究者就可以針對找出的使用者行為模式，進行網站版面的修改或是新增使用者有需求的功能，甚至可以在使用者的行為路徑中安排劇情式的廣告來達到強化網站效能的作用。以下我們列舉一些相關的文獻，並且說明我們的研究與其不同之處：

(1)最大向前參考序列的演算法：Chen 等人(1998)提出一個 MF(Maximal Forward)的演算法，將 Log Data 轉成最大向前參考序列，演算法強調 Web 上的應用，

要找出使用者瀏覽網頁的模式，但因預設使用者瀏覽網頁是不能有迴圈的。這和一般瞭解的網頁瀏覽行為較不相同。

(2)以停留網頁時間做門檻值的演算法：Hsieh 和 Chang (2001)提出一個 ITIM(Integrated Transaction Identification Module)演算法，主要是將使用者停留網頁的時間當作門檻值，來過濾出使用者有興趣網頁，進而求得最常瀏覽路徑。

(3)在多維資料庫中找尋序列模式的演算法：Yu 與 Chen(2002) 提出了一個在多維資料庫中找尋序列模式的演算法，這個演算法在每一階掃描中建立一個 Candidate tree 來做為計算 Support 值的依據。在每棵樹中不同的維度裏，該演算法都只採取一種的掃描方式來計算 Support 值，進而找出 Large Sequence 和隱藏的規則。

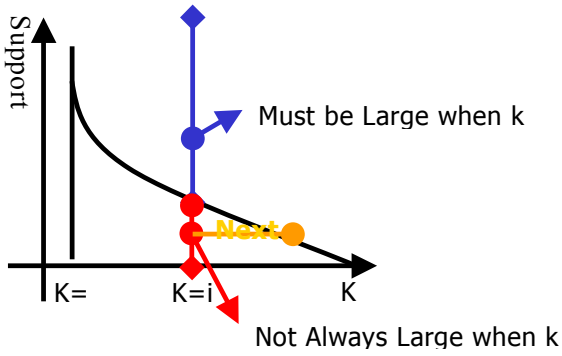
但因為這個演算法是建立在一般性的資料庫上，所以採取了只用一種掃描方式來處理 Support 值。在網頁-動作路徑情境中如果採用這個演算法，有可能會遺露掉一些較有趣的規則，如前後頁出現的網頁路徑中有什麼樣的使用者行為：而這遺露的規則對於想研究網頁瀏覽行為的人員來說，是相當的重要的。

(4)從 Log 檔中找出關聯規則及序列模式的演算法：Cooley 等人(1997b)之演算法計算 Support 的方式是看 Sequential Pattern 是否在一筆交易中出現，但是因為並未考慮到單筆交易可以出現重覆次數，及並未考量使用者在網頁中的動作，所可會容易忽略到使用者記錄中所更隱藏的內涵。即使如蘇育民（民 90）、丘文源（民 92）也只是猜測使用者可能的意圖行為，做處理網頁瀏覽路徑的權重，找出的規則仍只考慮到瀏覽的網頁而非加上動作。

(5)可重複序列挖掘網路瀏覽規則的演算法：陳仕昇等人(民 88)提出了 Next Pass Large Threshold 與 Next Pass Large Sequence 演算法，增進了一般瀏覽規則無法重覆計算 Support 的缺點。由於本研究重覆計算的方法是由此演算法改造而成，所以在這裏我們要仔細的介紹一下這個演算法的核心概念。可重複序列演算法的核心概念有二，一是門檻值的計算不同。一般不重覆計算的演算法，在門檻值的計算上，只需將資料總數乘上百分比型態的 Min.Support 即可。但是在重覆計算的演算法中，單一資料的 Support 數可能會大於 1，所以我們要先計算出所有可能 Support 值的總數，這個總數在每次掃描階數不同時(Pattern 不一樣長時)，皆會不同。算出總數後，乘上百分比型態的 Min.Support 就得到了門檻值了。第二個核心概念是往後看 X 步的想法，圖一說明瞭可能發生的狀況。

在普通的演算法中，只要當計算出的 Support 值小於門檻值，那就不能算 Large，在下一輪產生候選序列（Candidate Sequence，CS）時就沒辦法加入生成。但是在重覆計算的演算法中，這樣做可能會遺失掉一些可能的 Large 的 Pattern。假設我們計算一個 Pattern，在重覆計算的狀況下它的 Support 為 5，但是這一階的門檻值要 6 才會 Large，所以我們不會算這個 Pattern 為 Large。而下一階的門檻值可能降低成 3，而由剛剛不 Large 的 Pattern 所生成的下一階的 Pattern，就有可能 Support 會超過 3，應該被認為 Large 才是。為了儘量避免遺失資料，

所以第二個核心的想法就被引入使用，當一個 Pattrn 被掃描完之後，狀況不再是只有 Large 或不 Large 了。而是多了一個 NextPassX Large 的狀況，這說明瞭下 X 步，這個 Pattern 所生成的子序列是還可能 Large 的。所以，這類的 Pattern 雖然不能加入找尋規則的行列中(不算 Large)，但是卻可以加入生成下一階的 CS。陳仕昇等人提出重覆計算的想法，但是因為沒有考量到使用者動作的構面，所以還是無法找出與之相關的規則。



圖一：陳仕昇等重覆計算演算法往後看 X 步的概念

(6)將購買物品與瀏覽路徑結合的演算法：Yun 與 Chen(2000)提出一個 WTM(Web Transaction Mining)演算法，主要是將使用者在購物網站中所購買的物品與瀏覽路徑相結合，以求得使用者可能在哪些網頁購買特定商品。

(7)將時間與瀏覽網頁次數加入探勘法則中的演算法：Zhang 等人(2000)所提出的一篇文獻中，則是將使用者瀏覽網頁的次數與時間加入瀏覽序列探勘中，來預測使用者下一步可能點選的網頁，這樣可以將網頁預先下載至使用者電腦中，以增加網頁瀏覽的速度。但由於並未考量使用者行為，所以也無法找出與使用者行為相關的規則。

綜合來說，由於本研究討論的是一個過去較少人談到的模式，對於相關的論文很少有能力再多處理使用者動作路徑的部份，唯一可以採用的多維資料庫演算法，又因為缺少重覆計算能力，以及掃描方式不儘相同，跟本研究有些許相異之處。

三、網頁-動作路徑的資料挖掘演算法

為利於瞭解，整個演算法之基本觀念先簡略介紹如下：(1)先分別對有可能的 Page 與 Action 利用 PA_ScanN 做預掃，以產生 LS_0，這包含所有可能 Large 的 Page 與 Action Code。(2)再針對 LS_0 中，Page 與 Action 的所有可能組合的 PA 元素，產生 CS_1，利用 PA_ScanN 做掃描，完成 LS_1。(3)利用 Product 的方式，對 LS_1 與 LS_1，產生 CS_2，再利用 PA_ScanN 個別掃描，完成 LS_2。(4) 當 n> 3 後，利用 $LS_{(n-1)}$ (PA_Join) $LS_{(n-1)}$，產生可能的 $CS_{(n)}$，如果再沒有 Large 了，就停止掃描。如果還有 Large，就繼續下一階。

（一）網頁-動作路徑

本研究主要的分析目標是全球資訊網中使用者所瀏覽網頁及其在瀏覽網頁上所做的動作。所以本節主要要來介紹網頁-動作路徑的資料結構及其所擁有的一些特性。首先要說明的是，本文假設使用者所瀏覽的網頁及動作的記錄，可以藉由網頁系統的記錄程式、Cookie 或會員制等方式將其留存在系統端裡。

1. 網頁-動作路徑的資料結構

基本上，網頁-動作路徑資料是由二部份的 Log 所組成的。當瀏覽網頁的使用者進入網頁後，系統會給與這個使用者任務一個獨特的序列號，也就是表二、表三中的 Session_Id。這個序列號可用來判定那些網頁及動作的瀏覽記錄資料是屬於同一次瀏覽任務所產生的。表二是記錄使用者曾經瀏覽過那些網頁(Page_Id)、開始時間(StartTime)、結束時間(EndTime)的 Log。表三是記錄使用者在哪時間（ExeTime）曾經作過那些動作的記錄檔。經由比對記錄資料，我們可以將原始的記錄資料利用時間排列出所需的序列資料。以表二與表三中可見的資料為例，經本研究之演算法可以排列出<AbaCcabBb>這樣的序列資料，並將其儲存在瀏覽及動作路徑序列的資料庫中(如表四)。

表二：使用者瀏覽網頁路徑的 Log

Page_Log			
Session_Id	**Page_Id**	**StartTime**	**EndTime**
S001	A	2004/02/10 00:00	2004/02/10 00:10
S001	C	2004/02/10 00:11	2004/02/10 00:30
S001	B	2004/02/10 00:31	2004/02/10 00:50
…	…	…	…

表三：使用者瀏覽動作的 Log

Action_Log		
Session_Id	**Action_Id**	**ExeTime**
S001	b	2004/02/10 00:03
S001	a	2004/02/10 00:09
S001	c	2004/02/10 00:18
S001	a	2004/02/10 00:25
S001	b	2004/02/10 00:27
S001	b	2004/02/10 00:40
…	…	…

表四：瀏覽頁面及動作路徑序列的資料庫

MainSequence	
Session_Id	**PA_Sequence**
S001	AbaCcabBb
S024	AcBabCaEacFx
…	…

本研究的網頁動作序列（PA_Sequence）是不會出現<AbcAca>這樣的序列組合的。如果動作 bc 與 ca 都是在同一頁 A 所做，且他們是上下頁的關係的話，我們是會將其視為在 A 頁當中連續運作了 bcca 這些動作。以下我們會討論一些與 PA_Sequence 相關的規則：

PA_Sequence 是由 Page_Id 以及 Action_Id 所組成的。假設 P 是所有的 Page_Id 的集合，A 是所有 Action_Id 的集合，S 是所有可能的序列組合，則：

Rule 3.1.1 任一個$s \in S$，s 中不會存在沒有所屬頁碼概念的動作。
Rule 3.1.2 任一個$s \in S$，s 中不會存在完全沒有動作的頁碼。

也就是說，不可能會出現有如<aAbc>或是<ABca>這樣的序列。這二個規則我們也可以說成：當用頁碼來切割一個 PA_Sequence 的時候。一定能將這個 PA_Sequence 轉換成 $p_1 + A_1 + p_2 + A_2 + \ldots + p_n + A_n$ 這類的模式。其中 p_1、p_2、…p_n 屬於 P 中的一個 Element，而 A_1、A_2、…A_n 則是 A 中不為空值的子集合。這裏我們所談的+符號，代表的是直接連結(Concatenate)運算，也就是說<Aab> +<Cc>的結果即為<AabCc>。

2. 網頁-動作路徑的 Path 與長度

若不管 PA_Sequence 中所有的動作碼，則僅有使用者的網頁瀏覽路徑。假設我們的 PA_Sequence 的資料為 s，則 PAGE(s)就代表這個序列所瀏覽的網頁路徑。相反的，如果不管 PA_Sequence 中所有的網頁碼，則僅有使用者所執行動作的路徑，我們以 ACTION(s)來代表這個動作路徑。如前文所述，PAGE(s)並不會出現二個頁碼相同且相鄰的狀況，但 ACTION(s)卻有可能出現二個動作碼相同且相鄰的狀況。

因為 PA_Sequence 有網頁瀏覽路徑及動作路徑這二個維度，所以它會有多種的長度計數方式。一是由瀏覽網頁的路徑 PAGE(s)來計算，另一則是由動作路徑來計算 ACTION(s)。舉例來說，當$s \in S$，$s =< AabBacCba >$時，則$|s|_P$稱為 s 的網頁路徑長度，在此例中為 3；$|s|_A$稱為 s 的動作路徑長度，此例中為 6，$|s|_P + |s|_A$則稱為序列 s 的絕對長度，我們用這個$\|s\|$符號來表示，在這個例子中，s 的絕對長度為 9。

（二）、網頁-動作路徑演算法(Page-Action Algorithm)

我們的演算法與 Apriori(Agrawal and Srikant,1994)求關聯規則一樣，需要求取各個候選子序列(Candidate Sequence,CS)的 Support 值。再判斷那些候選子序列的 Support 值是否超過門檻值(Large Threshold)，藉以產生所謂的 Large Sequence(LS)。完成了 LS

之後，接著用 LS_k 產生 CS_{k+1}。得到 CS_{k+1} 之後，再重覆以上動作，直到不能再找出 LS 為止。本研究基於此種概念，發展出能配適在上節所述資料結構的新演算法。以下將演算法中的幾個重要元件與動作分別闡述。

1. 網頁-動作路徑的 Pattern (P-A Pattern)

在本研究中，Pattern 就是可能成為 Candidate Sequence 以及 Large Sequence 的元件。Pattern 與 Log 資料雖然一樣都是由頁面碼及動作碼所組成；但是 Pattern 的解釋及架構卻和 Log 資料有明顯的不同。

首先我們先定義一個 Pattern 的基本構成要素：

Definition 3.2.1.1 一個頁面碼加一個動作碼所組成的單元稱為 PA 元素。而 Pattern 則是由一或多個 PA 元素所組合而成的。

基於 Definition3.2.1.1 所說，我們可以很容易的知道一個 P-A 的 Pattern，他的絕對長度一定是偶數。依 Definition3.2.1.1 所定義的，[AaBb]可以是一個 Pattern，這個 Pattern 在本研究的解釋意義是指「使用者在 A 這一頁中曾執行過動作 a，且 A 頁瀏覽後下一頁是 B，在 B 頁曾執行過動作 b。」

一個 Pattern 的網頁路徑是指這個 Pattern 會走過的網頁瀏覽順序。如 Pattern [AaBbCc]中的[ABC]。由於 Pattern 與 PA_Sequence 不同，是可能出現[AaAb]這種模式的。所以要求出一個 Pattern 的網頁路徑，除了要去除動作碼之外，還要將相連重覆的網頁碼去除。一個 Pattern 的網頁路徑可以用來快速比對 Log，可用以加速演算過程。

2. 網頁-動作路徑的 Pattern 掃描(P-A_ScanX)

在我們產生候選子序列之後，要去計算整個資料庫，共有幾組滿足這個候選子序列所陳述的狀況。這就叫做掃描。一般 Sequence Pattern 掃描可分兩種，一種是只考慮 Element 是否與 Pattern 出現在序列的順序一樣，不考慮其間的連續出現性。以[ACE]這個 Pattern 為例，Sequence Pattern 中的掃描會認為序列<ABCDE>與<ACEBD>都是滿足[ACE]這個 Pattern。另一種掃描法，它考慮的就是這個 Pattern 必須完整連續出現在序列中，才算是滿足。同樣以[ACE]這個 Pattern 為例，在網頁瀏覽路徑的掃描中，<ACEBD>是滿足[ACE] Pattern 的，而<ABCDE>則不滿足。

連續出現的掃描模式，一般使用於只想找某個連續出現關係的時候。由於掃描要連續出現才算滿足，所以能找出的規則就只在連續出現的關係上。而不需連續出現的掃描模式，則可以找出某個序列先後關係的規則。但是，因為強調先後關係，對於連續出現的規則，就無法特別的指出來了。

在我們的研究裏，也面臨了網頁與動作二個維度該選用什麼掃描方式的問題。表五顯示了四種可能的掃描選擇。

表五：四種可能的掃描方式

網頁維度 / 動作維度	連續出現	前後出現
連續出現	Type Ⅰ	TypeⅡ
前後出現	TypeⅢ	TypeⅣ

在選擇要使用的掃描方法前，我們可以先分析一下這四種方式各會找出具有什麼解釋意義的規則。假設我們找到了一個 AbBc → AbBca 50%這樣的規則。如果這個規則是由 Type I 的掃描方式中找到，我們會將其解釋為「當使用者在 A 頁中做了 b 動作，且在下一頁 B 頁中做了 c 動作，使用者有 50%的機率在 B 頁的 c 動作做完後接著做 a 動作。」

如果規則是由 TypeⅡ的方式找到，我們會將規則解釋為「當使用者已在 B 頁前曾在 A 頁中做了 b 動作，之後在 B 頁做了 c 動作時，將有 50%的機率使用者會在做完 c 之後直接做 a 動作。」

如果規則是由 TypeⅢ的方式找到，我們則會將規則解釋為「當使用者已先在 A 頁中做了 b 動作，之後在 B 這一頁做了 c 動作時，將有 50%的機率使用者之後會在同頁做 a 動作。」

最後，如果規則是由 TypeⅣ的方式所找到，我們則會將規則解釋為「當使用者已在 B 頁前曾在 A 頁中做了 b 動作，之後在 B 頁做了 c 動作時，將有 50%的機率使用者之後會在 B 頁做 a 動作。」

理論上，這四種方法都是可以研究的。但是在目前本文所鎖定的範圍中，我們選擇了一個較合適也較有趣的方式來掃描。由於我們希望找出可提供重組網站結構或廣告配置參考的規則。在這個前提下，我們認為網頁路徑的掃描方法應以概念較固定、強調同時出現路徑的連續掃描較好。網頁採用「連續出現」的掃描方法，可以找出某一張網頁的上下頁關係，對於改善網站配置應較有幫助。也可以避免因為網頁瀏覽路徑過長，找到許多較無關緊要的先後關係。

另一方面針對現有網站架構與使用者一般瀏覽網站的行為看來，使用者在一張網頁中，可看到多個可實行的動作，使用者操作這些動作的自由度相對於網頁的瀏覽來的高的多。對於一張網頁中動作配置的討論，也大概只有方便與操作性，並不一定有重要的先後關係。另外以現在一般網頁的設計看來，使用者常常做了一二個動作，就可能被引導到其它的呈現頁去。每頁可收集到的動作數目，恐怕只會有二、三個。綜合以上兩點，在本文中，我們採用在概念上較寬的「不連續掃描」作法來處理動作路徑。除了較合乎動作在網頁中可能的不連續性外，也可以增加找出的規則。對於想瞭解使用者的行為，應較有幫助。

所以，本研究採取 TypeⅢ二種模式混合存在的掃描方式。在網頁元件(序列中屬於網頁代碼的元件)間的關係，我們採用「連續出現」的掃描法，而網頁中的動作元件

(序列中屬於動作代碼的元件)間的關係，我們則採用「不連續出現」的模式來掃描。在以下(第三小節)中，我們會用較類似真實例子的資料來說明，這樣的選擇應該是合理的。實際掃描的方式，我們用表六舉例說明。

表六：部份 MainSequence 資料庫的資料

Session_Id	PA_Sequence
S005	AacbBacAbc
S010	AaBcabCac
S015	BcaCabcAcb
S020	AcbCacbcaBca

假設我們產生了一個[AcBc]的 Pattern，這個 Pattern 的意思代表「在 A 這頁瀏覽時，使用者有做過動作 c，他下一頁瀏覽 B 頁，而且使用者做了 c 動作。」依照這樣的說法，由表五中，我們知道，只有 S005 滿足這樣的條件。值得一提的是，S020 中，雖然在 A 頁中使用者有做 c、在 B 頁中使用者也有做 c 的動作，但是因為這二頁中使用者還到過 C 頁，所以並不滿足我們的掃描法則，所以不算 Support 值。

另一種可能會產生的 Pattern 是[AcAb]這類的 Pattern，這類 Pattern 是由二個(以上)一樣的頁面代碼相鄰所產生的 Pattern。其意義為「同樣在 A 這一頁，使用者曾經先做過 c 這個動作再做 b 這個動作。」依此說法，S005、S015、S020 皆滿足這樣的掃描法則。

P-A_ScanX 依計算 Support 的方式不同，區分成 PA_Scan1 與 PA_ScanN 二種演算法，PA_Scan1 是指滿足條件的資料，不論單筆資料其中可滿足多少次的 Pattern 數，皆算一次 Support 值。而 PA_ScanN 的演算法，則可對滿足多次 Pattern 的資料，計算對應次數的 Support 值。例如：[AaBb]的 Pattern 在資料<AabBabCcAaBb>中，採用 PA_Scan1 只算一次的 Support，但是如果在 PA_ScanN 的演算規則之下，將計算二次的 Support 值。在這裏要特別提出來的是，如 Pattern：[Aa]在序列<AabaBacAac>中，採用 PA_ScanN 描掃法的話，Support 只會計算 2 次，因為[Aa]的解釋是在 A 頁中做了 a 動作的模式，在第一頁的序列中<Aaba>已滿足了在 A 頁中做了 a 的動作，雖然做了 a 的動作二次，但是在我們的研究中，不去計算動作部份的加權值。而最後一段的子序列<Aac>又滿足了一次 Pattern。所以這個任務的 Support 計算了 2 次。

以下是 PA_ScanN 掃描法的邏輯作法：

```
PA_Scan_N( Pat as Pattern, Seq as Sequence) as Integer{
  00   int i,j,k,Support = 0;
  01   int PPLen = Pat.Page.length; //PPLen=Pat 的網頁長度
  02   For (i=0 ;i<=(Seq.Page.length-PPLen);i++) {
  //對一個 Seq，一次比對 PPLen 長度的資料，所以共要比對(Seq 的網頁長度-PPLen)次
  03     If(Pat.Page==Seq.Page.substring(i,PPLen)){ //如果 PageCode 比對相符
```

```
04      int key=1
05      For (j=0;j<=PPLen-1;j++) { //再針對每個 Page 進去比對
06        int x=0;
07        String XPat=Pat.Action.atPage[j]; //XPat 是 Pat 中第 j 頁所做的動作
08        For (k=0;k<=XPat.length-1;k++){
09          x= Seq.Action.atPage[i+j].indexof(Xpat[k],x);
//x 代表這個 action code 第一次出現的位置，x 為-1 表示沒有出現這個 action code
10            If(x==-1){key = 0; j=PPLen; k=XPat.length;}
//沒有出現該 action code,這個比對失敗
11        }//End For Loop L08~L11
12    }//End For Loop L05~L12
13   If (key == 1) { Support++;} //當比對成功，Support 值+1
14   }//End If L03~L14
15   }//End For Loop L02~L15
16 Return Support;   //傳回 Support 值
17 }//End Function
```

PA_Scan1 掃描法的過程，與 PA_ScanN 類似，有差別的是將 PA_ScanN 中第 13 行加上一個跳出 02 比對 For 迴圈的 i=(Seq.Page.length-PPLen)。如果加上該行，則程式當發現此筆紀錄已計算過一次 support，就不會再比對下去。

3. 網頁-動作路徑的門檻值(Large Threshold in P-A Algorithm)

我們採取與傳統相同的方式，一個 Pattern 在資料庫中出現的次數要超過門檻值，我們才稱這個 Pattern 是 Large。對應於之前所說的二種 Scan 法，在訂定門檻值也有二種不同的做法。

若用 PA_Scan1(不重覆計次的掃描法)，我們採用與 Agrawal and Srikant (1994)中將門檻值訂為資料庫中資料總數的百分比值，這一 Large Threshold 是一個固定的數值。

若用 PA_ScanN(重覆計次的掃描法)，我們認定 Large 的方式與陳仕昇等人(民 88)採用的 Next Pass Large Threshold 與 Next Pass Large Sequence 類似。首先，我們得先對序列資料庫計算所有的可能 Support，再利用其所有可能的 Support 值來設定 Min.Support。在 PAScanN 這裏有一點必須要說明的，在我們的研究中，每一階掃描所產生出的 Pattern，雖然 PA 元素的數目都一樣，但是 Pattern 的網頁路徑長度並不會一樣。舉例來說，Pattern [AaBa]與[AaAc]都是可能在第二階掃描中產生出的 Pattern，但是[AaBa]是網頁路徑長度是 2，而[AaAc]的網頁路徑長度是 1。所以在我們的 PA_ScanN 演算法中，會發生同一階裏，會對應到不同的 Min. Support 值。為了避免過多不必要的計算，我們採用維護一個名叫 Total Possible Support 的資料庫，這個資料庫儲存著序列資料庫在特定的網頁路徑長度時，所擁有的所有可能 Support 數。這個最大的 Support 數乘上百分比型態的 Min. Support 值，就可做為特定網頁路徑長度的門檻值了。

4. 網頁-動作路徑的 Join (P-A_Join)

本研究的產生 Candidate Sequence 的方式採用類似 Srikant 與 Agrawal (1996)的 join 方法，也就是利用 LS_k join LS_k 產生 CS_{k+1}。我們稱 LS_k 為*原生 Pattern*；而稱 CS_{k+1} 為*生成 Pattern*。以下我們定義*原生 Pattern* 與*生成 Pattern* 的關係

Definition 3.2.4.1 X 與 Y 是二個原生 Pattern，如果 Z 是 X 與 Y 共同生成 ***Pattern****，則 Z 必然同時滿足 X 與 Y 的各別 Pattern。*

因為本研究所面臨的特殊問題，所以必須採用較特別的方法實作 Join 的動作。我們發現二個 Pattern 必須是可接合或融合的，才可以 Join 出生成 Pattern。以下我們就可接合與可融合的 Pattern 分項討論：

(1)一對可接合的 Pattern

當一個 Pattern 去掉頭一個 PA 元素(包含一個網頁碼及一個動作碼)，與另一個 Pattern 去掉末一個 PA 元素是完全相同時，我們稱這對 Pattern 是「一對可**接合**」的 Pattern。比如[AbBa]與[BaCc]的 Pattern 或是[AbBcBa]與[CcAbBc]，這些 Pattern 可以頭尾接合而產生如[AbBaCc]或是[CcAbBcBa]。這種比他原生 Pattern 多出一個 PA 元素的新 Pattern。這是 P-A_Join 生成下一階 CS 的第一種做法。

(2)一對可融合的 Pattern

P-A_Join 生成下一階 CS 的第二種做法，我們稱之為**融合**。這種情形則發生於一個 Pattern 的網頁路徑是另一個 Pattern 網頁路徑的子序列，而且這二個 Pattern 只有一個 PA 元素是不同的。就像是[AbBa]與[AcBa]的 Pattern 或是[AbCaBa]與[AbCbBa]這樣的 Pattern。這類型的 Pattern，我們處理的方式是融合，將相同的部份保留，然後在不同的位置上分別加入不同的部份。加完了之後，還要再考慮排列的問題。就像是第一對的 Pattern 會產生[AbAcBa]與[AcAbBa]這二個下一階的 CS。而第二組的二個 Pattern 則會產生[AbCaCbBa]與[AbCbCaBa]這二個 CS。

我們可以証明，任一個第三階以上的 Pattern，一定是由上一階的兩個父母 Pattern，由接合或融合這二種方法其中之一所生成的。

(3) P-A_Join 的過程

一開始系統會進行兩、兩比對的工作。比對完後就依類別進行接合或融合的動作，並將生成出的 Pattern 加入 CS_{k+1} 中。當所有的可能都掃描過後，還要清除 CS_{k+1} 中相同的資料，才算完成 P-A_Join 的工作。

整個 P-A_Join 的演算邏輯如下：

```
P-A_Join(LSList) as CSList {   //本函數需輸入 LSList，會產生 CSList
For i = 0 to LSList.Index.Count-1
  For j = i to LSList.Index.Count-1
    If (比對二 Pattern 是否只有一 PA 元素不同) {
      If (如果某一個 Pattern 的網頁路徑是另一個網頁路徑的子序列的話){
```

```
            進行融合程式;
            將融合結果加入 CSList;
        }
        Else If(不同的 PA 元素，一個在頭一個 PA 元素、
                另一個在末一個 PA 元素) {
            進行接合程式;
            將接合結果加入 CSList;
        }
      }
    Next j
  Next i
}
```

（三）一個可能真實例子的操作

在本節裏，我們將針對一個較實際的例子來實作演算法。為了能夠較方便的來表示較真實範例的網頁動作路徑。我們還是採用一些代號來表示可能的網頁碼與動碼。表七、表八是我們在這個例子中會使用到的基本代號。

表七：一個可能真實的 PageCode 與其代表

PageCode	代表頁面
Home	首頁
PInfo	商品資訊頁面
OnSale	特價品資訊頁面
Laws	法規資訊頁面
Submit	訂單確定頁面

表八：一個可能真實的 ActionCode 與其代表

PageCode	代表動作
H01	查詢最新消息
H02	登錄會員
P01	檢視詳細商品資訊
P02	查閱商品討論
O01	點選廣告資訊
O02	點選推薦品資訊
L01	查詢購物(交易)需知
L02	查詢會員需知
S01	將商品加入購物車
S02	將商品從購物車移除
X00	送出購物單，完成訂貨

在實際的情境中，我們假設使用者沒有做任何動作就會自動轉址的網頁是無效的。也就是說，對我們有意義的網頁，使用者至少要做一個動作。另外，在實務上，網站某網頁可執行的動作其實已經被網站經營者設定好了，系統能取得使用者的網頁與動作路徑是會在限定的網站地圖中的。了解一個網站其中的設計，可以有助於事前的資料清洗與事後的規則評估。表九是一段可能真實的網頁動作路徑資料。S001 使用者的 PA_Sequence「Home(H02、H01)→OnSale(O02、S01)→Submit(X01)」表示：使用者在系統首頁中先登錄會員，然後查詢最新消息的資訊。而後使用者跳到了特價品資訊頁，查詢了一個推薦的特價品資訊，之後他將商品加入購物車當中，最後跳到了訂單確定頁，送出購物單後完成訂貨的動作。

表九：一段可能真實的 MainSequence 資料

Session_Id	PA_Sequence: s	$\|s\|_p$
S001	Home(H02、H01)→OnSale(O02、O01、S01)→Submit(X01)	3
S002	Home(H01、H02)→PInfo(P01、P02、S01、P02)→Laws(L01)→Submit(S01、X01)	4
S003	Home(H01)→OnSale(O02、P01)	2
S004	Home(H02)→PInfo(P01、O01、P02、S01)→OnSale(O02、S01、O02、S01)→Submit(S02、X01)	4
S005	OnSale(O02、O01、S01)→Laws(L02)→Submit(S02)	3
S006	Home(H01)→OnSale(O01、O02、P01)→Laws(L01)	3
S007	Home(H02、O01)→OnSale(O02、O01、S01)→PInfo(P01、P02、S01)→Submit(S02)	4
S008	PInfo(P01、O01、P02)→Laws(L02、L01)→Home(O01、H01)	3
S009	Home(H02、O01、H01)→OnSale(O02、S01)→PInfo(P01、P02)	3
S010	Home(H02)→PInfo(P01、O02、S01、O01)→Submit(X01)	3

有了原始資料之後，我們就可針對這個資料進行網頁動作路徑的資料挖掘。在這裏我們採用的是 PAScan1 的演算法，門檻值設為 30%，超過 80%發生機率的規則，才算是常發生的。我們先採用預掃，來減低資料數量。預掃結果 L02(查詢會員需知)這個動作並未達到門檻需求。可不列入第一階 PA_Join 的處理單元中。

完成了預掃之後，就開始進入 PA_Scan1 各階的重覆演算中。表十、十一、十二為本範例產出的部份各階 Large Sequence。

表十：本範例部份的 Large Sequence 1 及其 Support 值：LS_1

Large Sequence	What	PA_Scan1 Support
Home(H02)	在首頁登錄會員	6
PInfo(P01)	在商品資訊頁檢視商品資訊	6
PInfo(S01)	在商品資訊頁將商品加入購物車	4

表十一：本範例部份的 Large Sequence 2 及其 Support 值：LS_2

Large Sequence	What	PA_Scan1 Support
Home(H02)PInfo(P01)	在首頁登錄會員後，接著去商品資訊頁檢視商品資訊	3
Home(H02)Pinfo(S01)	在首頁登錄會員後，接著去商品資訊頁將商品加入購物車	3
PInfo(P01)PInfo(S01)	在商品資訊頁中曾先檢視商品資訊，再將商品加入購物車	4

表十二：本範例部份的 Large Sequence 3 及其 Support 值：LS_3

Large Sequence	What	PA_Scan1 Support
Home(H02)PInfo(P01)PInfo(S01)	在首頁登錄會員後，接著來到商品資訊頁中，曾先檢視商品資訊，然後會將商品加入購物車	3

完成了各階的 LS 之後，接著就是找尋出滿足條件的規則。篇幅有限，表十三只列出發生機率超過 80%的部份規則，可概分為前導、後推、中夾型。將這些規則以人類了解的語言書寫後，可得表十五。

表十三：此例部份的發生機率超過門檻值(80%)的規則

編號	規則左邊（前提）	規則右邊（結論）	發生機率	規則類型
1	PInfo(S01)	PInfo(P01)PInfo(S01)	100%	←前導型
2	Home(H02)PInfo(P01)	Home(H02)PInfo(P01)PInfo(S01)	100%	→後推型
3	Home(H02)PInfo(S01)	Home(H02)PInfo(P01)PInfo(S01)	100%	→←中夾型

表十四：此例部份發生機率超過門檻值(80%)規則的解釋

編號	說明
1	已知一人[在商品資訊頁面曾將商品加入購物車]，有 100%的機率 他曾在同一頁中[檢視過商品資訊]
2	已知一人[在首頁曾登錄過會員，而且他接著在商品資訊頁執行過商品資訊的查詢]，有 100%的機率 他會[在商品資訊頁將商品加入購物車裏]
3	已知一人[在首頁中曾登錄會員，而且他接著在商品資訊頁將商品加入購物車裏]，有 100%的機率 他在加入購物車前[曾在商品資訊頁執行過商品資訊的查詢]

表十四整理出的規則，是隱藏在本研究範例資料中在依本研究演算法所能找到的規則。根據這些規則，網站的經營者可用改善網頁的配置或是廣告的位置。例如，經營者得到了規則 2 之後，可以針對在首頁登錄後，而又跳至特價品資訊頁的使用者，增加生動的廣告誘使他能夠去查詢商品資訊。這也許就能夠有效增加使用者將商品放至購物車的機率。這類規則是無法用一般只考慮網頁路徑演算法找出的。另外，我們觀察規則 1，這個規則可讓經營者知道，會將商品加入購物車的使用者，他會先做那些動作。但仔細觀查表九可以發現，沒有一筆資料是先查閱商品資訊後，下個動作就加入購物車的。如果對於動作路徑的掃描採用「連續出現」的模式，就無法找出這類的規則。

（四）本演算法的效率與效益

由於我們同時進行二種不同編碼的序列規則萃取，所以在找尋規則的每個步驟都較一般的網頁瀏覽規則演算法來的複雜。所以在執行效率上會較普通的網頁瀏覽規則來得低。(模擬測試相關的數據資訊，在第肆節會深入探討。)

但是就效益來談的話，本演算法可以找出網頁中與網頁間使用者行為動作的關聯規則，這是一般演算法所無法達到的，以下我們詳述本演算法可以找出的規則類型，及其應用：

一、常出現的使用者網頁-動作路徑：我們的演算法可以找出使用者常用的網頁路徑及其中使用者會進行的動作。網站的經營者可以藉由這樣的規則來調整選單的位置，或是在適當的位置設置適當的廣告，以提昇效益。在系統面，網站管理者如果知道這樣的資訊，就可以瞭解這個使用者的行為路徑是很重要的，所以與這個行為路徑有關的程式或是系統快取，應該特別加以優化。

二、網頁間的網頁-動作關聯規則：我們的演算法可以找出使用者執行到某一個階段時，其前推或是後推的網頁-動作會是什麼。也許我們會找出「當使用者在 商品資訊頁 執行 觀看付款規則 的動作時，有 80%的使用者下一步 會去 線上訂購頁 進行 下單訂購 的動作」這樣的規則。這種後推型的規則，網頁經營者可以預測使用者在其網站上的行為，在滿足某些瀏覽動作規則時，經營者可以安排出現某些會引吸使用者的消息或廣告，可以增加網站的效益。前推型的規則是指「已知使用者在 A 網頁做了 a 動作，那他可能有 70%的機率是由 B 網頁而來，在其中做過 b 這個動作。」在效益上，也是可以幫助網站經營者瞭解使用者的行為。

三、網頁中的網頁-動作關聯規則：我們的演算法也可以找出使用者在同一頁中所做動作的先後關聯。比方說，我們可以找出「當使用者在 商品資訊頁 先執行了 手動輸入機型 這個動作，他在同一頁還會在之後執行 將資料寄送給我 動作的機率為 80%」，網頁經營者瞭解了這樣的狀況，對於這兩個功能選單的網頁配置，就應該更注意；可能是要將這兩個功能選單放在附近，或是在使用者手動輸入了機型之後，就出現在資訊結果的旁邊。

參、雛形系統實作與執行效能分析

本研究實作了一個雛形系統，其中包括了整個網頁-路徑規則演算的流程。雛形系統是採用 Java J2EE 的 MVC 架構建置完成的。進行資料效能測試的機器是採用 Windows XP 的作業系統，CPU 為 AMD Athlon 2400+，主記憶體則是 256MB DDR 133。為了測試這個雛形系統的執行效能，我們還需要再定義如表十五所列的幾個參數，在測試過程中，我們將鎖定某些參數值來對特定的參數做資料測試。

表十五：測試資料有關參數

參數名稱	參數代號	所代表的內涵
資料庫數量	N	資料庫，所擁有資料的總數目。
Min.Support	MinS	本次掃描任務認定如何的資料才算是 Large 的門檻值。
平均網頁路徑長度	AVGPL	平均一筆資料的網頁路徑長度
平均動作路徑長度	AVGAL	平均一個網頁中的動作路徑長度
平均網頁路徑離散值	AVGPVar	資料庫中網頁路徑平均離散值
平均動作路徑離散值	AVGAVar	動作路徑的平均離散值

接下來的模擬與測試過程，我們將針對以下三大項目來討論這個雛形系統的效能：(1)資料庫型式、MinS 值相同，但資料量不同，(2) 資料庫型式、資料量相同，但 MinS 值不同，(3) MinS、資料量相同，但資料庫型式不同。其中，資料型式不同的問題又可以分為，網頁路徑平均長度不同、動作路徑平均長度不同，網頁路徑離散程度不同以及動作路徑離散程度不同四個方向。這裏所測量的時間數值，並未包含原始 Log 處理的時間。而是由使用者觸發進行 Scan 時開始計算，並於系統完成所有回應後停止計算。所有的計時工作，皆是由系統完成。

為了進行測試，需要產生大量符合各種參數需求的資料。本系統設計一個序列產生器產生 23 個資料庫來測試各種不同的狀況。

一、資料量不同對於網頁-動作路徑演算法的影響

在第一個模擬中，我們使用了編號 1 到 10 號的資料庫，MinS 設為 30%，其餘四項資料庫參數，皆為 AVGPL=3、AVGAL=4、AVGPVar=0、AVGAVar=0。圖二是使用 PAScan_1 演算法所呈現出的效能狀況。我們可以發現，PAScan1 的執行時間與資料總數具有一個近似線型關係。

PAScan_N 演算法因其 Next Pass 數目的不同所呈現出的效能狀況也有不同，當我們設定 MinS 為 25%時，掃描編號 1 到 10 的資料庫，我們得到圖三的數據，其代表 PAScan_N 在 Next Pass = 0 與 Next Pass = 1 兩種狀況所呈現出的效能狀況。

PAScan_N 在不同數量的資料庫中所執行出的效能看來也是呈線性關係的。而「不向後看」，與「向後看一步」這兩種演算方式在這 10 個資料庫的執行效能則是相差四倍左右。

二、Support 數不同對於網頁-動作路徑演算法的影響

針對同一個資料庫，MinS 數值越小，系統執行的效能應該會越慢，執行所需的時間就越多。為了驗証及觀察這個現象，我們在這次的模擬中使用了編號 11 的資料庫，資料量有 1000 筆，平均網頁路徑為 3，離散值 1，平均動作路徑為 4，離散值為 2。執行出來的數據如 圖四與圖五。由此二個圖看起來，不論是 PAScanN 或 PAScan1 系統的效能與 MinS 的大小成反比，而且皆是二次曲線的關係。

三、資料型態不同對於網頁-動作路徑演算法的影響

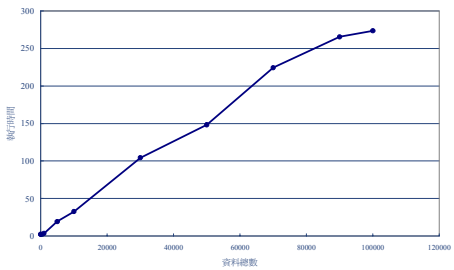
一個網頁-動作序列的資料型態可以用四個參數來測量。

平均網頁路徑長度　首先，我們使用編號第 12、13、14 三個資料庫，進行針對不同平均網頁路徑長度的資料庫做效能測試，這三個資料庫，平均動作路徑皆為 4，二項變異值皆為 0，在 PAScan1 演算法中，我們設定 MinS 為 30%，而 PAScanN 演算法中，我們設定 MinS 為 20%。圖六與圖七是測試的結果。我們可以發現，網頁路徑越長，所需要的執行時間就越長。二者之間，有正比的關係。

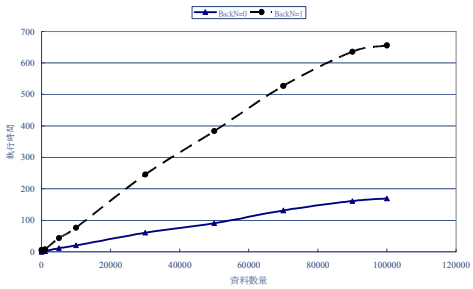
平均動作路徑長度　如果當我們使用編號第 15、16、17 三個資料庫，進行針對不同平均動作路徑長度的資料庫做效能測試，這三個資料庫，平均網頁路徑為 3，二項變異值皆為 0，在 PAScan1 演算法中，我們設定 MinS 為 20%。而 PAScanN 演算法中，MinS 則設為 10%。我們可以得到圖八與圖九的測試結果。也可以發現，動作路徑平均長度越長，所需要的執行時間也就越長，二者間，有正比關係。

平均網頁路徑離散度　如果針對兩個長度變異數值來做系統測試，我們得使用資料編號 18、19、20 與 21、22、23 兩組的資料庫集合。針對網頁長度路徑的離散數值所做的測試是在 PAScan1 的 MinS 設為 30%，PAScanN 的 MinS 設為 15%下測試的，我們可以發現如圖十與圖十一所示不管離散程度如何變化，對於效能並沒有直接的影響。

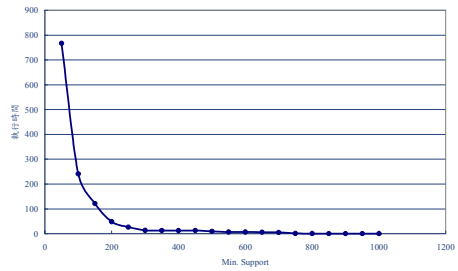
平均動作路徑離散度　我們在針對動作長度路徑離散數值所做的模擬測試是將 PAScan1 演算法中的 MinS 定為 30%，PAScanN 的 MinS 定為 15%，我們可以由圖十二、圖十三發現結果與上一個模擬的結果類似，動作路徑的離散程度並不明顯影響效能。



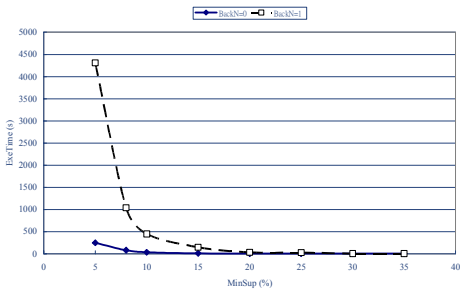
圖二：PAScan1 在不同資料量所需的執行時間



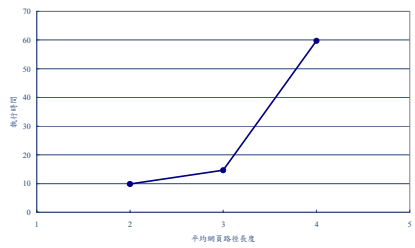
圖三：PAScanN 二種 Next Pass 參數在不同資料量所需的執行時間時間



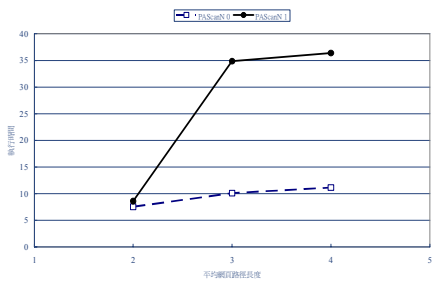
圖四：PAScan1 對同一資料庫在不同 MinS 所需的執行時間



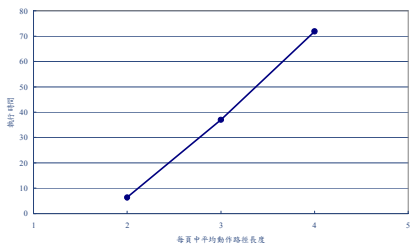
圖五：PAScanN 對同一資料庫在不同 MinS 所需的執行時間



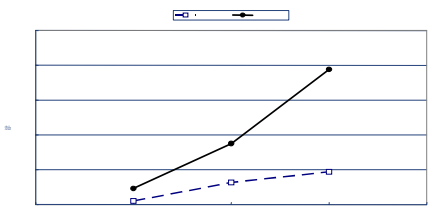
圖六：PAScan1 對不同平均網頁路徑長度所呈現的效能



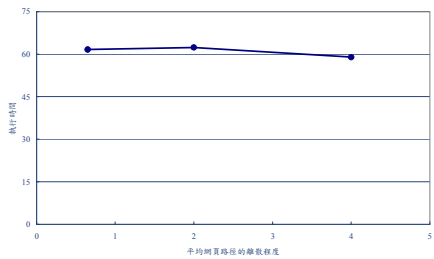
圖七：PAScanN 對不同平均網頁路徑長度所呈現的效能



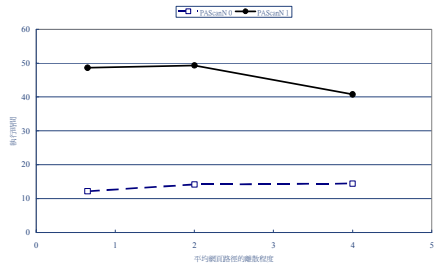
圖八：PAScan1 對不同平均動作路徑長度所呈現的效能



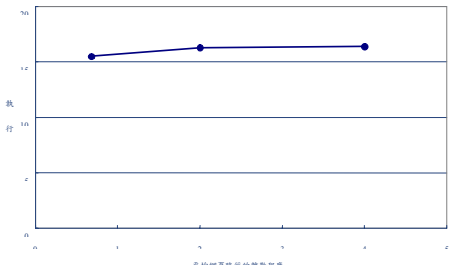
圖九：PAScanN 對不同平均動作路徑長度所呈現的效能



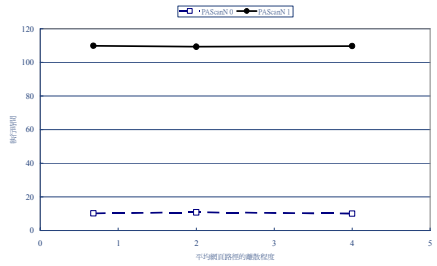
圖十：PAScan1 對不同網頁徑的離散程度的效能呈現



圖十一：PAScanN 對不同平均動作路徑長度所呈現的效能



圖十二：PAScan1 對不同動作路徑的離散程度的效能呈現



圖十三：PAScanN 對不同動作路徑的離散程度的效能呈現

肆、結論

一、研究貢獻

在全球資訊網上做使用者資料的資料挖掘主要目的就是要瞭解使用者在網站中的行為。在我們的研究中，我們對於觀察的資料顆粒相較於網頁瀏覽路徑的研究來的小，所以可以找出更多與使用者相關的動作規則。

（一）學術上的貢獻

在學術上，本研究首先發展出了一個同時將網頁-動作路徑一起考慮的關聯規則演算法 PA_ScanX。本研究對這類問題上的定義，可做為想研究找出這類問題其他規則的研究者做一個參考。

（二）實務上的貢獻

在實務上，我們建立了一個可以收集、處理網頁-動作路徑的網站模式。這可做為網站經營者意圖瞭解使用者行為模式時的一個參考模式。本研究找出的規則可以用在以下幾個方面：

(1)系統預測出使用者可能的要瀏覽的路徑，系統先將網頁資料暫存來增進系統效能之用。

(2)系統可以預測出使用者可以的瀏覽路徑，可以進行富有劇情式的廣告或行銷手段。

(3)系統可預測出使用者可能會有的行為路徑，經營者可以增強其客製行銷能力。例如當我們找出「當使用者在 A 這一頁中做了查詢規則的動作，他的下一頁將會在 B 頁下單的機率是 70%」這樣的規則時，經營者調整網站，使得當使用者在 A 頁執行了查詢規則之後，就可以自動展示與 B 頁有關的特價資訊。

二、後續研究建議

雖然本研究發展了一個針對網頁-動作路徑找出規則的演算法，並且也實作出了一個雛形系統，但是，未來還可繼續的研究包含：

(1)可將重覆計算的維度增進到動作路徑的維度：本研究目前尚無法處理同一個網頁中發生多次動作的重覆計算，主要遇到的困擾是如果將重覆計算的維度增進的動作路徑時，所有可能的 Support 總合(TPS)值可能會有隨著路徑長度遞增的狀況，這會破壞 Large 的假設，也就是在現在的路徑是 Large 的，但是在下一個路徑長度時，原本 Large 的 Pattern，就有可能不會 Large。這部份的問題還等待解決。

(2)一個更佳效能的演算法：本研究主要是採用傳統的 Apriori 的演算法來做修改，如同許多學者的討論，Apriori 原來的演算法效能不算很好，後續的研究者可以提出具有更佳執行效能的演算法來增強這類問題的處理效能。

(3)可將本研究的架構與系統實作與企業內：本研究目前還處於理論及驗証的階段，並未實際對使用者去收集或分析資料。後續的研究可以將可以在組織中實作系統，進行建置、收集、資料清洗、評估規則、應用規則等本研究尚未涵蓋到的方面。

參考文獻

1. 丘文源，含意圖行為之網路交易探勘演算法之整合研究，義守大學資訊工程學系碩士論文，民 92。
2. 陳仕昇、許秉瑜與陳彥良，「以可重複序列挖掘網路瀏覽規則之研究」，*資管評論*，第九期，民 88，頁 53 至 71。
3. 蘇育民，意圖行為於網路瀏覽習慣探勘之探索，義守大學資訊工程學系碩士論文，民 90。
4. Agrawal, R., Imielinski, T., and Swami A., “Mining Associations between Sets of Items in Massive Databases,” *Proc. of the ACM-SIGMOD 1993 Int'l Conference on*

Management of Data, Washington D.C., May 1993, 207-216.

5. Agrawal, R. and Srikant, R., "Fast Algorithms for Mining Association Rules," *Proc. of the 20th Int'l Conference on Very Large Databases*, Santiago, Chile, Sep. 1994.
6. Agrawal, R. and Srikant, R., "Mining Sequential Patterns," *Proc. of the Int'l Conference on Data Engineering (ICDE)*, Taipei, Taiwan, March 1995.
7. Baglioni, M., Ferrara, U., Romei, A., Ruggieri, S., and Turini, F., "Preprocessing and Mining Web Log Data for Web Personalization," *Lectures Notes in Computer Sciences*, Vol. 2829, September 2003, pp.237-249.
8. Batista, P. and Silva, M.J., "Mining Web Access Logs of an On-line Newspaper," *The 2nd International Conference on Adaptive Hypermedia and Adaptive Web Based Systems*, Malaga, Spain, May 2002.
9. Chen, M-S., Han, J. and Yu, P. S., "Data Mining：An Overview from a Database Perspective," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 8, No. 6, 1996, pp. 866-883.
10. Chen, M-S., Park J-S. and Yu, P. S., "Efficient Data Mining for Path Traversal Patterns," *IEEE Trans. on Knowledge and Data Engineering*, Vol. 10, No. 2, April 1998, pp. 209-221.
11. Cooley, R., Mobasher, B. and Srivastava, J., "Web Mining: Information and Pattern Discovery on the World Wide Web," in *Proceedings of the 9th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'97)*, Nov. 1997a.
12. Cooley, R., Mobasher, B. and Srivastava, J., "Grouping Web Page References into Transactions for Mining World Wide Web Browsing Patterns," *Proceedings of the 1997 IEEE Knowledge and Data Engineering Exchange Workshop (KDEX-97)*, Nov. 1997b.
13. Cooley, R., Mobasher, B. and Srivastava, J., "Data Preparation for Mining World Wide Web Browsing Patterns," *Journal of Knowledge and Information Systems*, Vol. 1, No. 1, 1999, pp. 5-32.
14. Frawley, W.J., Paitetsky-Shapiro, G. and Matheus C. J., "Knowledge Discovery in Databases: An Overview," *Knowledge Discovery in Databases*, California, Edited by Paitetsky-Shapiro, G. and Frawley,W.J., AAAI/MIT Express, 1991, pp.1-30.
15. Han, J., Cai, Y. and Cercone, N., "Attribute-Oriented Induction in Relational Databases," in G. Piatetsky-Shapiro and W. J. Frawley(eds.), Knowledge Discovery in Databases, AAAI/MIT Press, 1991, pp. 213-228.
16. Han, J., Cai, Y. and Cercone, N., "Knowledge Discovery in Databases : An Attribute-Oriented Approach," *Proceeding of the 18th VLDB Conferenc*e, Canada, August, 1992, pp. 547-549.
17. Han, J. and Fu, Y., "Discovery of Multiple-Level Association Rules from Large

Databases," *Proc. of 1995 Int'l Conf. on Very Large Data Bases (VLDB'95*), Zuich, Switzerland, September 1995, pp. 420-431.

18. Han, J., Yang, Q. and Kim, E., "Plan Mining by Divide-and-Conquer," *Proc. 1999 SIGMOD'99 Workshop on Research Issues on Data Mining and Knowledge Discovery (DMKD'99)* , Philadelphia, PA, May 1999, pp. 8:1-8:6
19. Hsieh, C. C. and Chang, C.T., "An Enhanced Transaction Identification Module on Web Usage Mining," *Asia Pacific Management*, pp.241~252, 2001.
20. Huan, J., Wang, W., and Prins, J., "Efficient Mining of Frequent Subgraphs in the Presence of Isomorphism," *Proceedings of the 3rd IEEE International Conference on Data Mining (ICDM)*, 2003, pp. 549-552.
21. Liu, J., Zhang, S., and Yang, J., "Characterizing Web Usage Regularities with Information Foraging Agents," Technology Report, COMP-03-001, Department of Computer Science, Hong Kong Baptist University, February 2003.
22. Nanopoulosa, A., Zakrzewiczb, M., Morzyb, T., and Manolopoulos, Y., "Efficient Storage and Querying of Sequential Patterns in Database Systems," *Information and Software Technology*, Vol. 45, 2003, pp. 23–34.
23. Pasquier, N., Bastide, Y., Taouil R. and Lakhal, L., "Efficient Mining Of Association Rules Using Closed Itemset Lattices," *Information Systems*, Vol. 24, No. 1, March 1999, pp. 25-46.
24. Pei, J., Han, J. Mortazavi-asl, B., and Zhu, H., "Mining Access Patterns Efficiently from Web Logs," *PAKDD'00 (Proc. Pacific-Asia Conference on Knowledge Discovery and Data Mining)*, Kyoto, Japan, April 2000
25. Pei, J., Han, J., Mortazavi-Asl, B., Wang, J., Pinto, H., Chen, Q., Dayal, U., Hsu, M., "Mining Sequential Patterns by Pattern-Growth: The PrefixSpan Approach," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 16, No. 10, October 2004, pp. 1-17.
26. Savasere, A., Omiecinski, E. and Navathe, S., "An Efficient Algorithm for Mining Association Rules in Large Databases," *Proc. Int'l Conf. Very Large Data Bases*, Zurich, Switzerland, Sep. 1995, pp. 432-444.
27. Srikant, R. and Agrawal, R., "Mining Sequential Patterns: Generalizations and Performance Improvements", *Proc. of the Fifth Int'l Conference on Extending Database Technology (EDBT)*, Avignon, France, March 1996.
28. Toivonen, H., "Sampling Large Databases For Association Rules," *The 22th International Conference on Very Large Databases (VLDB'96)*, Mumbay, India, Sep. 1996, pp. 134-145.
29. Yu, C-C., and Chen, Y-L., "Mining Sequential Patterns from Multi-dimensional Sequence Data," forthcoming in *IEEE Trans. On Knowledge and Data engineering*, 2005.

30. Yun, C.H. and Chen, M.S., “Using Pattern-join and Purchase-Combination for Mining Transaction Patterns in an Electronic Commerce Environment,” *The 24th Annual International Conference on Computer Software and Applications*, Taipei, Taiwan, pp.99~104, Oct 2000.
31. Zhang, W., Xu, B., Song, W., Yung, H. and Liu, K., "Data Mining Algorithms for Web Pre-fetching,” *Proceeding of the First International Conference on Web Information Systems Engineering*, Hong Kong, China, Vol.2 pp.34-38, June 2000.